Multimedia Integrated Modeling System

Table Of Contents

Introduction

What is a Framework?

A modeling framework application provides an interface so that users can design and augment projects that involve many programs and files. It gives users a common interface to access similar files, libraries, and executable programs. A framework connects these elements together and allows the user to ascertain the progress of multiple operations. By using the framework approach, investigators are encouraged to establish standard formats for sharing information.

The Multimedia Integrated Modeling System (MIMS) is a framework application that supports composing, configuring, and executing complex simulations involving multiple models and their associated tools (e.g., iterators and analysis tools). The MIMS framework directs the execution of models based on the user's designations of data dependencies.

The MIMS framework can enable compatible environmental models to communicate with one another, sharing input and output. As part of the MIMS design, users may pull models from different media (atmosphere, land, surface water, macrobiota, groundwater, etc.) into a single software application and enable users to study an entire ecosystem. By acting as a single framework, MIMS will promote the use of common data sets and fluxes to describe interactions at natural interfaces (air-surface water, surface water-groundwater, groundwater-macrobiota, etc.).

A MIMS Project serves as the user's library of building blocks (referred to as "members" in MIMS): data sets, executable programs, other tools, and operational sequences. The Project's members can be shared among both projects and users.

CAUTION: Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust. Be sure to review all modules, formulas for parameters, formulas in control file templates, and module preprocessors from outside sources before executing a scenario.

Why Should I Choose the Framework Approach?

The MIMS Framework gives you a flexible way to integrate different tools (models, analysis tools, gridding tools, iterators, data sets, etc.) by building scenarios. A MIMS Scenario will dictate how the tools communicate and how the tool parameters should be set. The ability to coordinate multiple models and tools is the central feature of the MIMS Framework. Through this coordination, you will have the ability to:

- Visually represent the data flow involved in complex modeling efforts
- Copy the specifications for complex model runs and then modify the runs
- Control multiple programs through a graphical user interface and eliminate the need to edit scripts and control files
- Perform complex model runs in a Microsoft Windows environment and share efforts with users on other platforms
- Interface some models that use different time steps or geospatial grids (future application)
- Integrate different models more easily
- Ensure that the correct input values are exchanged between models when repeating a simulation
- Automate repetitive tasks, such as Monte Carlo uncertainty studies
- Integrate data analysis tools to execute during and at the conclusion of a simulation
- Allow new users to follow the course of previous investigators

What Are Some Sample MIMS Projects?

The MIMS Framework will be a useful tool for multimedia and many other modeling studies. A few examples of current applications are highlighted below:

1. To predict ambient ozone concentrations, air quality modelers first need to know pollutant emission rates. The Sparse Matrix Operating Kernel Emissions (SMOKE) system is designed to estimate emissions. SMOKE represents a series of MIMS modules whose outputs are used as input for the Community-Scale Model of Air Quality (CMAQ) modules. The CMAQ modules then predict the ambient ozone concentrations.
2. To determine the optimal design for urban drainage systems, investigators are coupling the Stormwater Management Model (SWMM) with nonlinear optimization tools under the MIMS framework. The SWMM module is evaluated by the optimization tools, and optimization tools feed back the most cost-effective proposal into the next run of SWMM. The MIMS ties that link one module's output to another's input are called Connections.
3. Dry deposition is one process that transfers pollutants from the atmosphere to land and water bodies. The dry deposition velocity is calculated from measurements of ambient ozone and acids on filters by the Clean Air Status and Trends Network (CASTNet). The application of the MIMS Framework to the CASTNet analyses will make the CASTNet model testing easier in the future and allow investigators to process data from multiple sites more quickly. Under the MIMS Framework, a Scenario would describe the deposition to a particular area. In many cases, a MIMS Domain Object represents a media (e.g., atmosphere, soil, surface water, and plants) or media interface (e.g., atmosphere-to-surface water) where a process or behavior occurs, and the MIMS modules are associated with a particular domain object.

Publications describing the applications and design of the MIMS Framework can be found at http://www.epa.gov/asmdnerl/mims/publications/index.html.

How Do I Install MIMS?

The MIMS Framework is designed to operate in the Java Runtime Environment (version 1.3 or higher). Below you will find instructions on:

Checking your current version of the Java Runtime Environment in Windows

Installing MIMS for Windows Systems

Installing MIMS for UNIX Systems

Checking your current version of the Java Runtime Environment in Windows:

Using the Windows Start menu:

1. Select Find

2. Select the File or Folder menu option

3. Search for java.exe on your local drives. You should check the box next to "Include subfolders" and then hit the "Find Now" button.
If you find the file, use a DOS prompt to put yourself in the directory the file is in. If you have more than one copy, check the one in the Windows or WINNT directory first.
If, for example, java.exe is in the folder c:\Windows\system\, you would type

```
> cd windows\system
```

at a command prompt (i.e., a DOS window). Note that in DOS, file names must be reduced to 8 characters or fewer.

Then type:

```
> java -version
```

at the command prompt.

Installing MIMS for Windows Systems:

1. Install the Java 1.3 or 1.4 Java Runtime Environment (for non-developers) or Java Development Kit, available for free from Sun at http://java.sun.com/j2se/. Note that some problems have been observed with Java 1.4.1 on Windows XP operating systems.

2. Download the MIMS Framework installation file from http://www.epa.gov/asmdnerl/mims/software/index.html.

3. Execute the downloaded zip file (mims_rel.exe) or run PKUNZIP on (mims_rel.zip) to extract the MIMS files.

4. Edit the batch file bin\runMIMS.bat to set the MIMSHOME variable to the directory where you installed MIMS (default is c:\mims).

5. Put Java 1.3 (or higher) on your path (using system properties on NT; this won't take effect until after you reboot) or edit runMIMS.bat to explicitly refer to the Java executable. The installations from Sun usually contain a bin directory, which contains the Java executable.

6. On Windows 95/98/Me, set the memory properties of the runMIMS.bat file so the initial environment has at least 2560 bytes.

7. Double-click runMIMS.bat or its shortcut (Windows 98/Me) to start the MIMS framework.

8. See the Demo QuickStart for an example using MIMS.

Installing MIMS for UNIX Systems:

1. Have your system administrator install the Java 1.3 or 1.4 Java Runtime Environment (for non-developers) or Java Development Kit, available for free from Sun at http://java.sun.com/j2se/.

2. Download the MIMS installation file from http://www.epa.gov/asmdnerl/mims/software/index.html.

3. Expand and untar the downloaded compressed tar file.

4. Edit the shell script file bin/runMIMS.sh to set the MIMSHOME variable to the directory where you installed MIMS.

5. Put Java 1.3 (or higher) on your path or set the JAVA_HOME environment variable.

6. Execute the runMIMS.sh to start the MIMS framework.

7. An example scenario is included and limited on-line help is available.

For help within MIMS, click on the Help menu and choose User Guide.

What are the First Concepts to Learn about the MIMS Framework Structure?

The MIMS Framework contains several hierarchical levels with which a user must become familiar before constructing a full project. After designing a project once, most users will be familiar enough with the framework structure to navigate through the MIMS windows.

Following the chapter on the QuickStart tutorial, the User's Manual presents a chapter on Terms and Concepts within the MIMS framework. That chapter provides more details on the concepts described below and highlights additional terms.

A MIMS Project serves as the user's library of building blocks (referred to as "members" in MIMS): data sets, executable programs, other tools, and operational sequences. Within a MIMS Project, users can define the members, their file paths, and their associated parameters. All of this information is then available for a user to create different simulations.

A MIMS Scenario defines the operational sequence and settings for a given simulation. The MIMS Scenario Window lets users follow the progress of a simulation as each member's operation is executed.

In the examples presented below, scenarios are usually composed of MIMS Domain Objects. A domain object is a grouping of processes, modules, and parameters that describe one area of a simulation. The most common domain objects represent a particular media (e.g., atmosphere, soil, surface water, plants, and exposed populations) or a natural interface (e.g., atmosphere-to-surface water) when the domain object describes fluxes between media. A scenario will generally contain one or more domain objects.

Within each domain object, one or more actions may occur, and the actions are defined as MIMS Processes. In most examples, the MIMS Processes will involve an action by the computer (running code, reading/writing data, displaying information, etc.). A Process will often require input information and generate output.

Many MIMS Processes are implemented by executing code, a MIMS Module. A MIMS module is essentially the device that would be operated to accomplish a particular task. An internal module may be implemented with Java code that is called by MIMS, and an external module represents a stand-alone program that may be invoked by MIMS. A single application of a Module to implement a MIMS Process within a Scenario is termed a Module Instance.

Scenarios, Domain Objects, Processes, and other MIMS members may have MIMS Parameters associated with them. Parameters refer to the information that a MIMS process needs to complete the simulation.

If the output from one process serves as the input to another, then users must define a Connection within the scenario. Connections allow users to track how one process feeds into another. In MIMS, the users do not define the sequence of operations explicitly. Instead, MIMS determines the order of events based on the connections.

In the Introduction, you're introduced to several events to illustrate how modules, processes, and domain objects fit together. The table below may be used as a reference to recall the nomenclature.

Event	Module	Process	Domain Object
CMAQ predicts ozone concentrations in the atmosphere.	CMAQ	Predicts ozone concentrations	Atmosphere
The rake creates piles in the yard.	Rake	Create piles	Yard
The bags transport leaves in the yard.	Bags	Transport leaves	Yard
The cream treats blisters as a first aid measure.	Cream	Treats blisters	First aid (framework object)
The dep_module simulates dry deposition in the atmosphere.	Dep_module	Simulates dry deposition	Atmosphere
The Time Series Plotter graphs the time series for the analysis.	Time Series Plotter	Graphs the time series	Analysis

Publications describing the design and applications of the MIMS Framework can be found at http://www.epa.gov/asmdnerl/mims/publications/index.html.

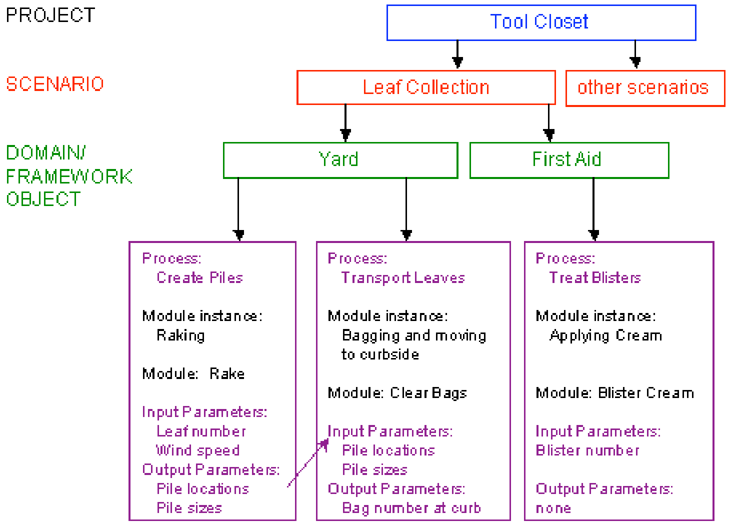
Is There an Analogy to Help Me Understand the MIMS Framework Structure?

A simple analogy for the framework would be to consider a MIMS Project as a tool closet. An organized tool closet may have different shelves for yard work tools, painting applicators, plumbing supplies, etc.

Most shelves may be regarded as MIMS Domain Objects, having only one place (media) where their tools would be useful. One would use a rake from the Yard work shelf outdoors but never consider using a rake in the kitchen. Domain objects are typically associated with a single media. Different tools may be available to accomplish the same task (e.g., both a rake and a leaf blower), and any could be used in the Yard.

The rake found on the yard work shelf would be analogous to a MIMS module. The rake is a device that can be used to accomplish a particular task. In MIMS, an individual task is described as a process.

The Create Piles process is associated with the rake as a tool. Use of a leaf blower would also fall under the Create Piles process, but the tool (module) would change. In MIMS, the rake creating piles in the yard one time would be defined as a module instance. The module instance may be influenced by different parameters (e.g., the number of leaves, the wind speed, and the desired location for the piles). In MIMS, Parameters are associated with every module instance. In MIMS, users would define the input parameters (number of leaves and wind speed) or specify a file to read. The output parameters (leaf pile locations and sizes) would be the results from the module instance.



After raking, the leaves must be bagged and transported to the curb. This particular sequence of events gives the homeowner the desired result and cannot be performed in any other order. Within MIMS, module instances are performed in a particular order as defined by the MIMS Scenario. The scenario approach lays out all of the tasks that must be performed and their sequence. For tasks more complicated than raking, the homeowner may choose to keep a copy of a sequential checklist in the tool closet, much the same way that MIMS users store scenarios within a MIMS Project.

The output from the Create Piles process was the pile of leaves, and this pile represented the input to the Transport Leaves process. Within MIMS, this relationship between the output of one module and the input for another is represented by a Connection. The sequence of the connections determines the order that calculations for a scenario should be performed.

Finally, some shelves in the tool closet contain items that may be used in different places but still have a focused purpose. For example, a first aid supply shelf in the tool closet may contain the blister cream that needs to be applied after the raking is completed. The first aid shelf would represent a MIMS Framework Object, and the blister cream container would be another MIMS module. Application of the blister cream would constitute a MIMS process. MIMS framework objects could include modules that are scheduling programs, Monte Carlo tools, visualization tools, and analytical programs. Framework objects generally do not represent processes within the simulation but instead support the simulation or its analysis.

How Are Parameters Associated?

In the Introduction's analogy, the Leaf Pile Locations parameter is an output parameter from the Raking module instance and an input to the Bagging module instance. Since the leaf piles are also associated with the Yard and the Leaf Collection, the Leaf Pile Locations parameter could have been considered a parameter of the Yard domain object and the Leaf Collection scenario.

The MIMS framework allows parameters to be passed between members within a scenario through four functions:

Button	Function Name	Function
⊞	Add to Scenario	Copy a parameter from a module or object within a scenario into the scenario parameters table
Right click	Add Parameter Source	Allow the parameter value from one object to be used in another object
⊶	Connect to Parameter	Allow the parameter value from one object or module to be used in another module or in the scenario
⌖	Copy from Module	Copy a parameter from a module within an object into the object's parameters table

When building parameter connections, it is helpful to understand how the variables are passed through a scenario after the connections are made. The following values are passed when the user executes the scenario (for the case where the module instance is associated with that object's processes):

- If an *input* parameter value is specified in the scenario parameters table and is connected to an object or to a module instance, the object or the module instance parameter assumes that value.
- Similarly, if an *input* parameter value is specified in the object parameters table and is connected to a module instance, the module instance parameter assumes that value.
- If an *output* variable is specified at the conclusion of a module instance and is connected to an object or scenario, the object or scenario parameter assumes that value.
- If an *output* variable is specified by an object and is connected to the scenario, the scenario parameter assumes that value.

The Source column of the parameter tables lets users know the source (module, object, and scenario names) of their parameter values. Many module instances will also need to draw parameters from objects other than their own. The

Connect to Parameter button allows users to feed input parameters to a module directly from the output parameters of other objects and from the scenario.

At the end of the QuickStart tutorial, users will encounter an example where the Dep_Monte scenario is nested as a module within the Scen_Monte Carlo scenario. In this case, the Scen_Monte Carlo scenario exchanges the parameters listed in the Dep_Monte scenario parameters table as the input and output. The outer scenario cannot read any of the other parameters within Dep_Monte.

What is the Goal of the Example Shown in the QuickStart Tutorial?

To familiarize you with some of MIMS's concepts and capabilities, a simple demonstration simulation has been included that uses a dry deposition model. This demonstration is likely not as extensive as a real application of the framework would be but is sufficient to illustrate the most important concepts. Only a Windows version of the model has been provided with this distribution.

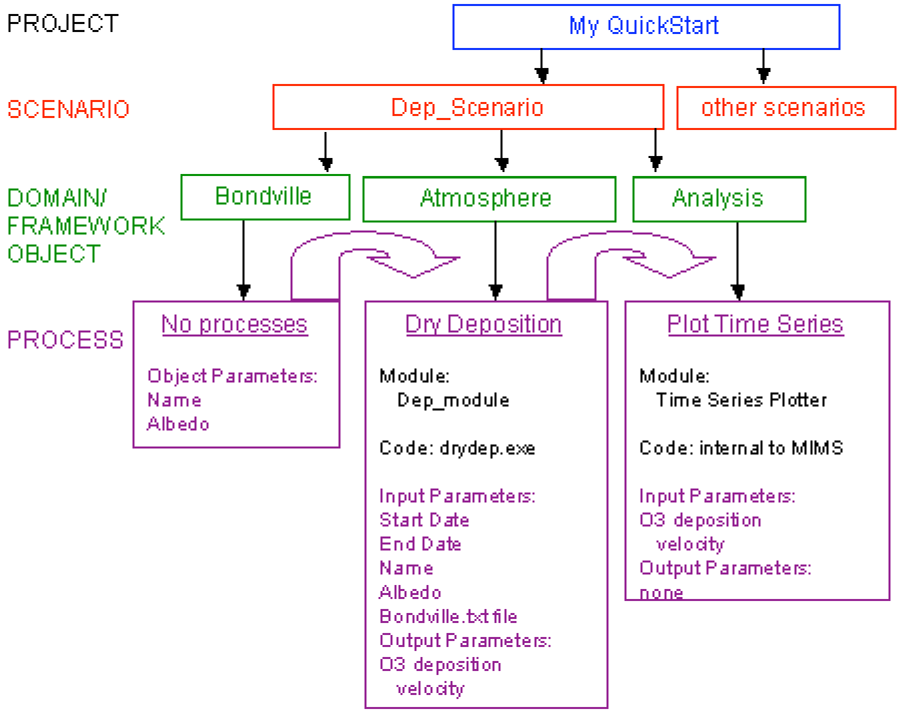
The dry deposition model computes the deposition velocity for selected trace gases. The deposition velocity is proportional to the flux (mass per unit area per time increment) of a chemical to the land surface. The deposition velocity depends on the state of the atmosphere, the plant canopy, the soil, and chemical properties.

The model that is used is a development version that incorporates the Multilayer Model (Meyers, T.P. et al. J. Geophys. Res., 103(D7), 22,645-22,661, 1998) and the Multilayer Bio-Chemical Dry Deposition Model (Wu Y., B. Brashers, P. L. Finkelstein, and J. E. Pleim, A Multilayer Bio-Chemical Dry Deposition Model, submitted to J.Geophys. Res.). The model and its data sets are in the QuickStart directory within the MIMS directory you installed. The model reads and writes data in a tab-delimited format, as seen in the file bondville.txt in the data subdirectory. The model also reads a text file that describes what computations should be performed. An example of this file is bond.input. It contains start and end dates, file names, control flags, and parameter values.

In this framework application, you will assemble the members of a MIMS Project called “My QuickStart.” Two MIMS Processes will be created to accomplish the tasks:

1. Dry Deposition: this process calculates the ozone deposition rates using the external dry deposition module
2. Plot Time Series: this process will display the calculated deposition rates versus time by using the internal module called Time Series Plotter

The diagram below illustrates the connections and nesting of the project members. Straight arrows connect to the members associated with a particular project, scenario, or object. The U-turn arrows show that parameters are passed between domain objects and into the next module instance.



The Bondville domain object will hold some input parameters but will not execute any processes. The Dry Deposition process will be located in the Atmosphere Domain Object and have the type Atmosphere. The Analysis process will appear in the Analysis Framework Object and have the Analysis type.

You are interested in the data contained by the Bondville Domain Object serving as the input to the Dry Deposition process, so a connection will be established in that direction. After the deposition fluxes are calculated, the Time Series Plotter will display the results. Therefore, a second connection will be required that carries output from the Dry Deposition process to the Time Series Plotter. This entire sequence will be controlled from the Dep_Scenario scenario.

After you create a Time Series plot, the QuickStart tutorial goes onto describe a way to repeat the scenario thirty times as part of a Monte Carlo simulation. The Monte Carlo simulation considers the uncertainty of plant cover values and shows how the maximum ozone deposition velocity changes with different values.

A sample MIMS project called Demo QuickStart has been provided to show what the completed example should look like. If you open the "deposition" scenario in that project, set the scenario parameter Default Directory to point to the QuickStart directory on your machine, and reset the scenario (all of those steps are described in the following chapter), you should be able to execute the scenario in the Demo QuickStart project.

How Do I Complete the QuickStart Tutorial?

How Do I Complete the QuickStart Tutorial?

The programs for the QuickStart Tutorial are currently only available for Microsoft Windows users. However, other users can also learn the methods for creating their own MIMS projects by following this guide.

The MIMS Framework opens a new window for every MIMS Project, Scenario, Domain Object, Framework Object, Module, and Module Instance. In this tutorial, you will become accustomed to creating, organizing, and dismissing these windows.

Since MIMS records the project information whenever you close a window, it is not necessary to save your progress as you proceed. However, to prevent loss of your work from a power or system failure, you should choose Save Project in the File menu after any significant effort.

In this tutorial, you will begin a new project. First you will add the deposition model to the project as a new module and then assign that module to a new domain object. You will create a second domain object describing the site data and then connect the two together with a scenario. You will execute the scenario and then add another domain object to display your results.

To begin MIMS, click on the MIMS icon to run the runMIMS.bat file. The MIMS Project Selection Window will appear.

Create the My QuickStart Project (QuickStart)

Under the File Menu on the MIMS Project Selection Window, select the New Project menu item. When prompted for the Project name, type “My QuickStart.” The MIMS Project Window will appear and show the contents of your project. The Category list near the top of the window shows the member categories in the project and is currently empty for this new project.

Create the Dep_Module MIMS Module (QuickStart)

You will first create a deposition module because MIMS includes a feature allowing you to copy information from a module into a domain object. From the My QuickStart window, click on the New button.

The Create Member dialog will appear. Select “Default External Module” from the Modules folder, indicating that you plan to use a program external to the MIMS framework. Type “Dep_Module” as the name for the module and then click the Create button.

The module window will then appear, and you must enter the module commands, classification, description, parameters, etc. Beside Execution Command, type “bin\drydep.exe –b config.txt” to describe the partial directory path (bin\), program name (drydep.exe), and command line arguments (-b config.txt). You will define the full default directory path when you create the scenario.

Beside Control Files, click the Add button. The MIMS control file acts as a template for the model’s control file. When prompted, name the control file config.txt.

Then click the Edit button on the Control Files line of the Module Dep_Module window. This button will open the Control File Editor . Instead of typing the entire control file for the drydep.exe program, this tutorial provides users with the file bond.input in the QuickStart directory. Open bond.input using Microsoft WordPad and copy all of the text in that file (Ctrl-C will copy the selected text).

Paste the text (using Ctrl-V) into the text area at the bottom of the Control File Editor. The text in the bond.input file was set up for a specific simulation, but you will modify the text to read the parameters from the MIMS windows. To read the parameters from within the framework, you will need to use the MIMS parameter substitution syntax (described further under Calling Parameters).

On the Control File Editor, replace the text “08/17/94” with “${getDate(“Start Date”,”MM/dd/yy”)}” and note that the command is case-sensitive. The getDate method will return the value of a date parameter as a formatted string. Next replace “10/01/94” with “${getDate(“End Date”,”MM/dd/yy”)}” to instruct MIMS how to insert the end date into the control file.

Next scroll down in the Control File Editor to the File Locations and Names Section of the config.txt file. In the first line, replace "bond" with "${str("Name")}" beside the title Station Name. Change the second line to read "${str("State")}   Input Data File, blank to use Sonic/Campbell data" by replacing the word "bondville." Also change the output data file line by typing "data\output.txt" in place of "data\bond_v25.txt."

The first thirteen lines should now read:

```
========= Input file for program DRYDEP.  Spaces don't matter. =====
25        Input file version number
```

```
\        Character used in filename parsing, "/" for unix, "\" for Windows
${getDate("Start Date", "MM/dd/yy")}  Start Date
${getDate("End Date", "MM/dd/yy")}  End Date
-15341 DATETIME Offset: 18993 for Statistica & Quattro, -15341 for jday (94)
BAB      Initials of operator
Comments (entire next line):
Berry model, output only good O3 data
=== File locations and names section =======================
${str("Name")}          Station Name
${str("State")}     Input Data File, blank to use Sonic/Campbell data
data\output.txt      Output Data File
```

Next scroll down to the last line of the User's Switches Section. Replace the text "0.10" with "${str("Albedo")}" to instruct the control file to accept the Albedo parameter.

Click the OK button to dismiss the Control File Editor .

The compatible domain object type determines which domain objects will be able to use this module. In the blank beside Compatible Domain Object Type, enter "Atmosphere."

The Process Name will be displayed in the Scenario Window and indicates the action represented by the module. In the blank beside "Name of Process This Implements," type "Dry Deposition."

Finally the Dep_Module module will need some parameters.

Adding Parameters to the Dep_Module Module (QuickStart)

Six parameters will be added to the Dep_Module module. To add a new parameter, you should click on the + or select Add New Parameter from the Parameters menu. A pair of dialogs will then prompt you for the parameter type and its name. These dialog boxes are followed by a dialog box that lets you set the parameter's attributes. After you choose the appropriate attributes, click the OK button at the bottom of this dialog and the Description dialog and add the next parameter.

The table below shows the parameters that you need to set up for the QuickStart tutorial (leave the domain object names and descriptions blank and the Is ReadOnly boxes unchecked for all parameters):

Type	Name	Domain Object Type	Input	Output	Required
Date parameter	Start Date	<leave blank>	Y(es)	N(o)	Y
Date parameter	End Date	<leave blank>	Y	N	Y
String parameter	Name	Site	Y	N	Y
Floating point parameter	Albedo*	Site	Y	N	Y
File parameter	State	Atmosphere	Y	N	Y
Floating point time series (generic)	O3 Deposition Velocity**	Atmosphere	N	Y	N

* Albedo units should read "na" since albedo is dimensionless. This is set in the Edit Albedo window that appears after setting up the Albedo parameter.

** Set the O3 Deposition Velocity units to "cm/s." This is set in the Edit O3 Deposition Velocity window that appears after setting up the O3 Deposition Velocity parameter.

If you make a mistake while creating the parameters, choose Undo from the Edit Menu. You may also delete the parameter through the ✕ button and begin again. Changes cannot be made to the above attributes after the parameter is created.

After creating the O3 Deposition Velocity variable, the Additional Type Info field will contain a button called Update. Click on the Update button. A dialog will pop up prompting you to choose the type of time series; choose "Time Series from File Value" and then click the OK button.

Then a dialog appears that allows you to edit the time series parameter. Beside File Name, type "${str("Default Directory")}\data\output.txt" to specify the name and location of the output file. Choose EST (Eastern Standard Time) as the time zone and check the box beside Apply Daylight Savings Time Adjustment.

The remaining information describes the file format. Leave the delimiter as tab, and uncheck the box beside Treat Multiple Delimiter As One. Enter "date+hour" in the Date Label field and type "MM/dd/yy HHmm" in the Format field (this format is not in the pulldown menu). Enter "mo3dv" in the Value Label field to indicate the column label for the O3 deposition velocity. Then click the OK button.

Close the Dep_Module window by choosing Close Module under the File Menu.

Create the Atmosphere MIMS Domain Object (QuickStart)

In the My QuickStart window, click New, select Domain Object, enter "Atmosphere" as the name, and click Create.

A domain object window opens. Enter "Atmosphere" as the type. From the Parameters menu select Copy from Module and choose Dep_Module. This action will copy into the domain object all relevant parameters (State and O3 Deposition Velocity) from the Dep_Module module instance.

To specify the domain object's processes, click the Processes tab then click Add Process and enter "Dry Deposition."

Close the Atmosphere Domain Object window by clicking on the X in the upper right corner of the window.

Create the Bondville MIMS Domain Object (QuickStart)

In the My QuickStart window, click New, select Domain Object, enter "Bondville" as the name, and click Create.

The Bondville Domain Object window will then appear. In the Bondville domain object window, enter "Site" as the type. Under the Parameters menu, choose Copy from Module and select Dep_Module. This action will add the parameters Name and Albedo to the Bondville domain object.

No processes are associated with the Bondville domain object.

Close the Bondville Domain Object window.

Create the Dep_Scenario MIMS Scenario (QuickStart)

In the My QuickStart window, click the New button, select Scenario as the type, enter "Dep_Scenario" as the name, and click the Create button. A window for the scenario will open. This window provides three views of the scenario's contents. On the left is an outline listing the scenario's domain objects in a tree view. The right side of the window shows a diagram with a graphical view of the scenario's domain objects and their connections. The table at the scenario window's bottom shows the scenario parameters.

Add a domain object to the scenario by right-clicking in the graph view. A menu should appear. Choose Add Domain Object. Select Bondville. A box with the name Bondville will appear where the mouse was clicked. Left-click on a domain object and hold the button down to drag the domain object to the upper left corner of the graph view.

Right click on an empty spot in the graph view. When the menu appears, choose Add Domain Object again and select Atmosphere. Drag the Atmosphere domain object to the area just below the Bondville domain object.

Connect the Domain Objects in the Scenario (QuickStart)

In the graph view of the Dep_Scenario scenario window, right-click on Atmosphere and choose Add ParameterSource from the menu. Then move the mouse to Bondville and left-click. A line should appear between the two domain objects with the arrow pointing to Atmosphere. This arrow indicates that the Bondville domain object provides parameters to Atmosphere and its processes.

Expand the domain objects by clicking on Expand All in the Object menu of the graph view. The Expand All command shows the processes in each domain object. To attach a module to implement dry deposition, right-click on Dry Deposition (in either the tree or graph views) and choose Set Module Instance. A list of the applicable modules will appear (based on domain object type and the process). Choose Dep_Module.

A dialog appears offering to connect some of the model's input parameters. Click the Create Connections button. A module instance is created that represents this invocation of the model.

In the scenario window, the Dry Deposition process now has a broken circle on its left and the phrase Dep_Module on its right.

Assign Values to the Parameters (QuickStart)

In the Dep_Scenario window, the partial circle to the left of the Dry Deposition process indicates that some information required to run Dep_Module is missing.

Double-click the Dry Deposition process to see the parameters in the module instance, which initially are copies of the parameters in the module. The red text in the Status column on the right indicate parameters that must be set. Set the following values by double-clicking in the Value column and entering the value:

Parameter	Value
Start Date	08/21/1994 00:00:00
End Date	08/26/1994 00:00:00

To use the model's result, select the O3 Deposition Velocity row and choose Connect to Parameter from the Parameters menu. A list shows the scenario and domain objects that could provide the parameter. Select Atmosphere and click OK. The next window shows the potential connections, with all selected by default. Press Create Connections. This indicates that the output of the model should be passed to the Atmosphere domain object. Note that the status column for O3 Deposition Velocity parameter indicates that it is used.

The other input parameters cannot be set here because they come from domain objects. The values must be set by those domain objects. Close the Deposition Model module instance window.

In the Dep_Scenario window, open the Bondville domain object window by double-clicking on its box. Enter "bondville" and "0.1" in the Value column for the Name and Albedo. Close the Bondville domain object window.

Open the Atmosphere domain object window by double-clicking on the word Atmosphere. Enter "data\bondville.txt" for the State value. Close the Atmosphere domain object window.

In the Dep_Scenario window there should now be an open green circle to the left of the Dry Deposition process. This green circle indicates that the module is ready to run.

Before we can run the scenario, we need to indicate the default directory for this work. Look at the Scenario Parameters table at the bottom of the Dep_Scenario window (if the panel is not showing, click on the small triangle at the bottom left of the window). Beside the File Management Parameters parameter, click the Edit button.

A new dialog will appear showing the twelve variables in the group called File Management Parameters. In the Default Directory row of this new dialog, click the Browse button. Find the QuickStart subdirectory within the MIMS directory, select it in the file list, and click the Select button. Close the dialog titled Parameter Values of File Management Parameters.

Execute the Scenario (QuickStart)

To execute the model, choose Execute All from the Scenario menu. The Dep_Module will be executed; as it operates, a filled green circle will appear beside the Dry Deposition process. A check mark in the green circle indicates that the execution completed. Some models indicate failure by returning a status code that MIMS recognizes. The demonstration model does not return such a status code, the Dep_Module always appears to complete successfully.

To see if the model did execute successfully, right-click on the Dry Deposition process and choose View Output Log. Microsoft Notepad will start and show the log file created by the model (if not, set the viewer to Notepad in the Edit Administration Information Window). A log file for a successful execution should be similar to the following text:

```
Reading config.txt
Reading bondvill.lai
Reading plants\PLANT.DAT
Reading plants\PADPROF1.20
Reading plants\PADPROF2.20
Reading plants\PADPROF3.20
Reading plants\LANG.SPH
Running Dry Deposition Model for bondvill
Output will be to data\output.txt
All data will be output, with bad values blanked.
Week 1: 08/21/94 08/22/94 08/23/94 08/24/94 08/25/94 08/26/94
Done: 288 records in 8.0 seconds: 36.0 records/second
```

If the scenario has been run by a previous user, the following line will also appear in the output log:

```
WARNING: Output File already exists and will be overwritten.
```

Close the Notepad window after examining the file.

Add the Analysis MIMS Domain Object (QuickStart)

We will use a time series plot to view the results. In the My QuickStart project window, click New, select Time Series Plotter from the Modules folder, name it "Time Series Plotter," and click the Create button. This opens a window for a predefined plot module. Close the window when it appears.

We will use a domain object to hold the plot routine. In a future version, a framework object will be available for this purpose. The tasks of framework objects are typically independent of the environmental models and instead represent computational functions. In the My QuickStart project window, click New, select Domain Object, name it "Analysis," and click Create.

Set the type to "Analysis" and add a Process called "Plot Time Series." Close the Analysis domain object window.

In the Dep_Scenario window, right-click on the background in the graph view. Select Add Domain Object and choose Analysis. This process brings the Analysis domain object into the scenario.

Right click on the Analysis domain object in the graph view, choose Add Parameter Source, and left-click on the Atmosphere domain object. Click on the "+" in Analysis to see its process. Right-click on Plot Time Series, use the Set Module Instance command on the right-click menu to choose Time Series Plotter, and click OK.

A dialog will appear with a list of possible parameter connections. The MIMS Time Series Plotter can plot ten variables simultaneously so multiple possible connections are shown. Click on the top row to highlight only that row then click Create Connections. The time series plotter will automatically use the output specified by the Atmosphere domain object.

In the Dep_Scenario window, right-click on the Plot Time Series process and select Update Module Instance. If you do not do this, the parameters in the output file will not be properly read by the Time Series Plotter.

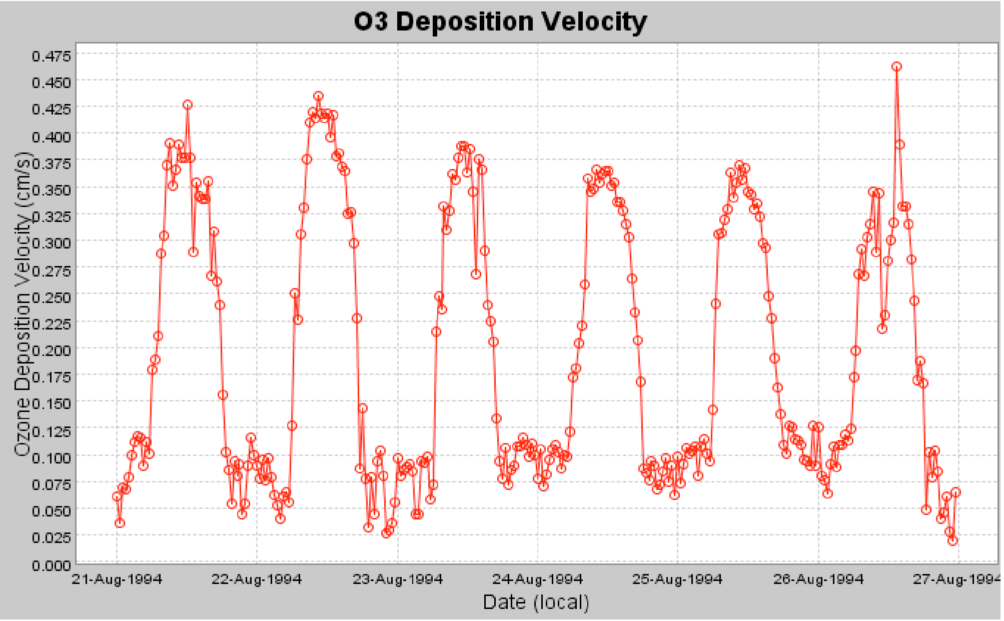
In the Scenario menu of the Dep_Scenario window, choose Reset All then Execute All. When the execution completes, you should see a time series plot of the deposition velocities.

Interpret the Results Using the Time Series Plotter (QuickStart)

The window plotting your modeled results is titled Analysis Time Series. The x-axis shows the dates from August 21-27, 1994 (the dates are centered on midnight), and the y-axis shows deposition velocities in cm/s. The plot shows the model output. Note that the ozone deposition velocities reach their maxima during the day and drop at night.

You can customize the appearance of the axes, labels, and background by right-clicking the mouse on the window and selecting Properties. Click on Properties. On the Chart Properties dialog, select the Plot tab and the Vertical Axis tab within that. Change the label from "cm/s" to "Ozone Deposition Velocity (cm/s)" and click the OK button.

Right-click again on the plot and select Print. Choose the appropriate printer in landscape mode and click the OK button. Your printed chart should look like this.



Congratulations on completing the QuickStart tutorial!

Now close the plot window.

Expand to Employ a Monte Carlo Approach (QuickStart)

In this expansion, you will add a new parameter to the Dep_Scenario scenario and then copy the scenario to a simulation called Dep_Monte. The Analysis object will be removed from the Dep_Monte scenario, in order to allow the Dep_Monte scenario to act as a single module within another scenario (Scen_Monte Carlo).

The Scen_Monte Carlo scenario will use a Monte Carlo Framework Object to feed the new parameter to the Dep_Monte module and will read back the maximum ozone deposition velocity. The output from the framework object will be displayed by a histogram plotter.

Buttons and menu commands will be listed in brackets to remind users of the necessary keystrokes.

Step 1. Create the Dep_Monte Scenario

From the My QuickStart project window, open the Dep_Module module window [select Dep_Module from the Default External Module list and click Open]. Add a "Primary Plant Percentage Cover" floating point parameter [+]. It should be assigned a domain object type of Atmosphere, as input, as required, and with units of %. This parameter represents the fraction of the land covered by the primary plant type and is often subject to some uncertainty.

From the Dep_Module module window, edit the config.txt control file [Edit button]. Near the end of the file, replace the explicit number 50 beside "Primary Plant Percentage Cover" with ${str("Primary Plant Percentage Cover")}. Change the Secondary Plant Percentage Cover from 50 to 0. Click on the OK button to accept the change. Then close the Dep_Module module window.

From the My QuickStart project window, open the Atmosphere domain object [select Atmosphere from the Domain Object list and click Open]. In the Atmosphere domain object window, click on the Copy from Module button ⚙. Choose the Dep_Module when prompted for the module name. This will add

Primary Plant Percentage Cover to Atmosphere's parameter list. Close the Atmosphere domain object window.

In the Dep_Scenario window, you'll have to delete the existing Atmosphere domain object [right click on Atmosphere and choose Delete Object]. Then add the new Atmosphere domain object [right click in an open area of the Graph View and choose Add Domain Object]. Set the module instance to Dep_Module [left click on the + sign in the Atmosphere box, right click on Dry Deposition, and choose Set Module].

Open the Atmosphere domain object window [double click on Atmosphere in the Graph View], and set the value for the State to "data\bondville.txt". Double click on the Dry Deposition process in the domain object window to open the module instance window. Set the values for the start and end dates to "08/21/1994 00:00:00" and "08/22/1994 00:00:00" in the module instance window. Close the module instance window.

In the scenario window, reattach the parameter connection to the Bondville domain object [right click on the Atmosphere domain object in the graph view, choose Add ParameterSource, drag the arrow tail to the Bondville domain object, select both Albedo and Name, and click Create Connections]. Next reattach the parameter connection to the Analysis domain object [right click on the Analysis domain object in the graph view, choose Add ParameterSource, drag the arrow tail to the Atmosphere domain object, select only the first variable, and click Create Connections].

From within the Atmosphere domain object window (opened within the scenario window), highlight the Primary Plant Percentage Cover parameter and then click the Add to Scenario button ⊞. Choose *Input* and Create Connection to Parameter in the Scenario Parameter Properties Dialog.

Do the same for the O3 Deposition Velocity parameter--choose *Output* and Create Connection to Parameter in the Scenario Parameter Properties Dialog. The scenario parameters table in the scenario window should now show the Primary Plant Percentage Cover with a Used, In status and the O3 Deposition Velocity parameter with a source of Atmosphere and status as Out. Type the value "70" into the value column for Primary Plant Percentage Cover in the Scenario Parameters table. Execute the scenario to be sure that the model still runs properly and produces an appropriate time series (two daily ozone peaks).

Close the Dep_Scenario and Atmosphere windows.

From the My QuickStart project window, highlight the Dep_Scenario and click on the Duplicate button. Name the new scenario Dep_Monte. Then open the Dep_Monte scenario window. In the Dep_Monte scenario window, right-click on the Analysis domain object and delete it. The Dep_Monte scenario will act as the module that operates repeatedly for the Monte Carlo simulation.

The Scenario Parameter Table should appear at the bottom of the scenario window (if not, select its checkbox under the View Menu). Click on the Edit button beside the File Management Parameters parameter in the Scenario Parameters table. This should open the window called Parameter Values of File Management Parameters. In this new window, select the Default Directory parameter and then click on the Remove from Group button (⊖) or select Remove from Group in the Parameters menu. This action transfers the Default Directory parameter up to the Scenario Parameters table.

Close the window called Parameter Values of File Management Parameters.

Close the Dep_Monte scenario window.

Expand to Employ a Monte Carlo Approach (QuickStart)

Step 2. Create the Monte Carlo Iterator

The Simple Monte Carlo Iterator (max) will send a single Monte Carlo parameter into a module (at the scenario parameters level) and output the maximum value for the output parameter for each Monte Carlo simulation. Create a Simple Monte Carlo Iterator (Max) in the My QuickStart project window named Monte Carlo [New button and found in the Iterators folder]. The Framework Object window will appear for the iterator. Click on the Processes tab. Highlight the Iterate module, and press the Edit Module Instance button to make the module instance window appear.

From the Module Instance Iterator window, click on "Set Iterand" in the Parameters menu. Choose the Dep_Monte scenario as the member. A Dep_Monte (Inputs/Outputs) dialog will appear. On the Dep_Monte Inputs tab, select Primary Plant Percentage Cover in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. On the Outputs tab, select O3 Deposition Velocity in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. Click the OK button. The dialog closes. Six new parameters are added to the parameters list: Execution Method (value of Local Execution), Random Number Seed, Default Directory, Number of Realizations, Distribution of Primary Plant Percentage Cover, and Realizations of O3 Deposition Velocity.

Set the value for the number of realizations to 30. Click the Set button beside the Distribution of Primary Plant Percentage Cover parameter. Choose a normal distribution (named Land) with mean of 50 and standard deviation of 10. After choosing the standard deviation, click on the mean to set the value and then click the OK button. Minimize the Module Instance Iterator window.

Expand to Employ a Monte Carlo Approach (QuickStart)

Step 3. Create the Histogram Plotter

From the My QuickStart window, click New and select Histogram Plotter from the Modules folder and name it Histogram Plotter (the compatible domain object type should be "Analysis"). Then create a new domain object [New button] named Histogram Analysis with type Analysis. Click on its processes tab and add a process called Plot Histogram to this domain object.

Minimize the Histogram Plotter, Histogram Analysis, and Monte Carlo windows.

Expand to Employ a Monte Carlo Approach (QuickStart)

Step 4. Create the Scen_Monte Carlo Scenario

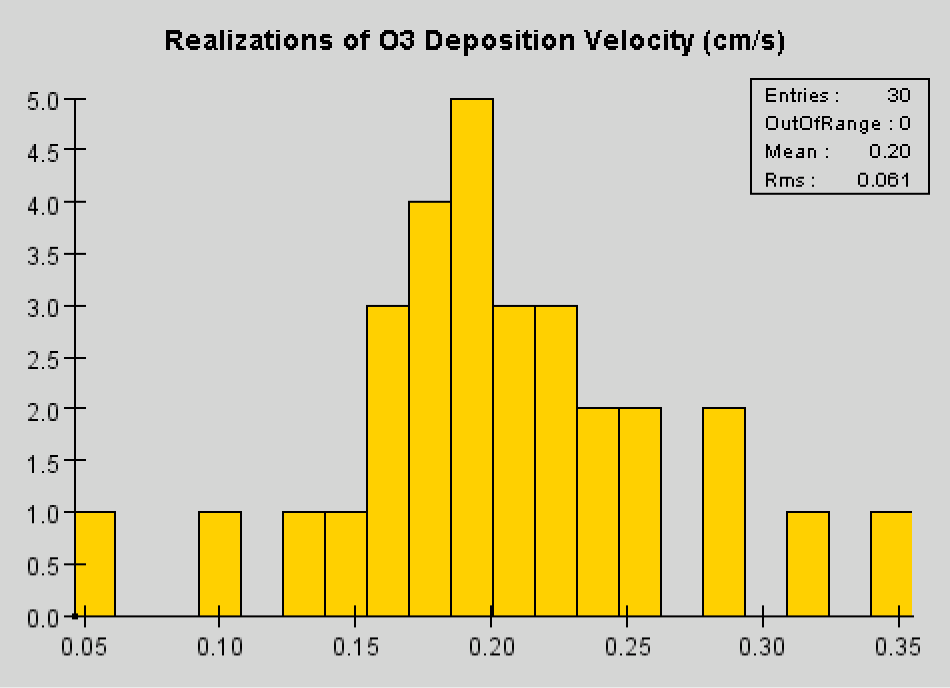
Create a new scenario named Scen_Monte Carlo for the Monte Carlo study [New button on My QuickStart window]. Set the default directory [first click Edit on the File Management Parameter group] to match the QuickStart directory used for the Dep_Scenario scenario [click the Browse button beside the default directory parameter]. [Close the dialog showing the File Management Parameter group].

Add the Monte Carlo framework object to the scenario [right click in the graph view of the scenario window and choose Add Framework Object]. An incomplete circle should appear, and this indicates that the Default Directory parameter of the Iterator process has not yet been set.

To set the Default Directory parameter to match the default directory for the Scen_Monte Carlo scenario, click on the plus sign in the Monte Carlo object. Double click on the Iterate process to create a window titled "Module Instance [Iterator] in Scenario [Scen_Monte Carlo]. In this new window, click on the Default Directory parameter and then on the Connect to Parameter button (-▪▪-). Select Scen_Monte Carlo in the dialog that appears and prompts you for the connection, and click the OK button. A new dialog appears, and you should click on the Create Connections button. The Default Directory for the nested scenario and the Scen_Monte Carlo scenario will now be identical. Close the module instance window, and you will see an open green circle beside the Monte Carlo framework object.

Also add the Histogram Analysis domain object. Click on the + sign in the Histogram Analysis domain object. For the process Plot Histogram, set the module to Histogram Plotter [right click on Plot Histogram and choose Set Module]. Right click on the Histogram Analysis domain object, click Add ParameterSource, and drag the arrow down to the Monte Carlo framework object. When the Select Connections dialog appears, only highlight the first row and click the Create Connections button.

Choose Execute All from the Scenario menu and examine the histogram plot that is created. Right click the mouse on the plot, and uncheck the box beside Show Error Bars. Then the plot should appear similar to this:



To produce precisely the same histogram, set the value for the Random Number Seed parameter in the Scen_Monte Carlo scenario to 5. Then execute the scenario again.

Terms and Concepts

Chemical Mechanism

A Chemical Mechanism describes all of the chemical and physical processes and rates for all chemical species. The chemical processes include chemical reactions for both the species of interest and tracer species.

The Chemical Mechanism module currently in MIMS focuses on atmospheric processes. Therefore, the chemical reactions (and reaction rates) may be described for the aerosol (solid), aqueous (liquid), and gaseous phases.

Physical processes are also described within the Chemical Mechanism because each chemical species may have different settling velocities, rates of conversion to other phases (e.g., from gas to aerosol), wet deposition rates, or surrogate species to be used for tracking.

Chemical Mechanisms are viewed and altered through the Edit Dialog for Mechanism.

Chemical Mechanism Family

To quantify the sensitivity that model results have to particular reactions or the reaction rate constants, research investigators may choose to vary the Chemical Mechanism. This approach is also useful to test the effect that simplifying assumptions have on the model results. For a particular MIMS Scenario, the MIMS Chemical Mechanism Family may be specified as a Parameter within a MIMS Module.

The Chemical Mechanism Family is a MIMS classification that allows investigators to group together similar chemical mechanisms, perhaps with just a single variable being altered at a time. The Chemical Mechanism Family Editor is the control window that allows users to examine and edit the family members.

Domain Object (presented in the Domain Object Window)

A Domain Object represents a grouping of MIMS Processes, MIMS Modules, and Parameters that all share a common theme or area of interest within a simulation. The most common domain objects are individual media (Atmosphere, Land, Lake Superior, Groundwater, etc.), but domain objects also often represent interfaces (Atmosphere-Land, Groundwater-Humans, etc.). In the My QuickStart Tutorial, a domain object is also used to represent the Analysis area of the simulation.

A MIMS Scenario is made up of one or more Domain Objects and Framework Objects. The Domain and Framework Objects represent the third hierarchical level of the MIMS framework (under projects and scenarios). Each domain object holds information describing its MIMS Processes, MIMS Modules, and Parameters. Each domain object may hold one or more processes that are used to describe the functions within the object's area of interest within the simulation. All parameters that are output from the domain object's processes are also described at the object level, and input parameters may also be specified at the object level.

In the Scenario Window, the Domain Objects are displayed. The Domain Objects are shown as subunits on the Tree View section of this window and as boxes on the Graph View section. Domain objects can be created directly from the Project Window. The specifications about domain object type, the domain object parameters, and the values can be set from the domain object window opened from the Project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when the user opens the domain object window from within the scenario, but the other specifications may not be altered.

Sample Domain Object: Atmosphere		
Sample Modules	**Sample Processes**	**Sample Parameters**
MM5 meteorology preprocessor	Calculate meteorology	Initial conditions Boundary conditions
CMAQ air quality model	Calculate air quality	Daily emissions Receptor latitude, longitude, and elevation Output file name

When a domain object is created, the user specifies its type. Only modules with that same type may be included in that domain object.

Embedded Analysis Modules

The MIMS Framework provides some simple analysis tools that allow users to plot and print their output information. These modules are easy to distribute along with the framework but will not have all of the functionality of the external analysis modules.

To use one of the embedded analysis modules, first create the module from the project window by selecting the desired plot type (histogram, time series plotter, etc.). Then create a domain object with a type of "Analysis" from the project window. In the domain object window, click on the processes tab, add a process, and set the process's module instance to the desired plot type.

Users may choose from the following embedded analysis modules and see them described further below:

Histogram Plotter

Time Series Plotter

The ***Histogram Plotter*** is a pre-programmed module within MIMS. The module will create a screen showing up to four histograms simultaneously. It implements the Plot Histogram process and has an Analysis object type.

In the parameters table, users must set at least the first series of data through a parameter connection to another object. Other parameters that can be specified in the module instance window are:

- Number of Bins -- Select the number of bins for the data. If the number of bins is not specified, the program will determine a reasonable default number.
- Tile Histograms? -- When the box is checked, multiple histograms will appear on the same page of the plot window. When the box is not checked, each histogram appears on its own tab.
- Series 2, 3, and 4 -- Show other histograms on the same plot window.

When the scenario is executed, a histogram plot window will appear. Users may right-click on the plot to change the error bars, colors, axis labels, titles, etc.

Right-clicking on the plot also presents the choices to print or save the image to a file.

The ***Time Series Plotter*** is a pre-programmed module within MIMS. The module will create a screen showing up to ten time series simultaneously. It implements the Plot Time Series process and has an Analysis object type.

In the parameters table, users must set at least the Time Series 1 through a parameter connection to another object. Other parameters that can be specified in the module instance window are:

- Window Title -- Set the title for the window. This does not set the title for the plot but just the window that holds it.
- Time Series 2 and higher -- Show other time series on the same plot window.

When the scenario is executed, a time series plot window will appear. Users may right-click on the plot to change the ranges, colors, axis labels, titles, etc. Right-clicking on the plot also presents the choices to print or save the image to a file.

External Analysis Modules

The MIMS Framework provides some complex analysis tools that allow users to plot and print their output information. Some external analysis modules will allow direct output (no user interaction) to graphic files and printers. These external analysis modules are distributed separately from the framework (unlike the embedded analysis modules).

To use one of the embedded analysis modules, first create the module from the project window by selecting the desired plot type (e.g., Plot Page Object). Then create a domain object with a type of "Analysis Engine" from the project window. In the domain object window, click on the processes tab, add a process, and set the process's module instance to the desired plot type.

Users may choose from the following external analysis modules and see them described further below:

Plot Page Object

The external analysis modules will be described further in the next MIMS documentation release.

Framework Object (presented in the Framework Object Window)

A Framework Object represents a grouping of MIMS Processes, MIMS Modules, and Parameters that all share a common functionality within a simulation. The tasks of framework objects are typically independent of the environmental models and instead represent computational functions, such as advancing time steps or repeatedly conducting a process.

A MIMS Scenario is made up of one or more Domain Objects and Framework Objects. The Domain and Framework Objects represent the third hierarchical level of the MIMS framework. Each framework object (within a Scenario) holds all of the information describing its Processes, Modules, and Parameters. Each framework object may hold one or more processes that are used to describe the functions within that grouping of the simulation.

In the Scenario Window, the MIMS Framework Objects are illustrated. The Framework Objects are shown as subunits on the Tree View section of this window and as boxes on the Graph View section.

The following Framework Objects have been created for use in MIMS:

- Iterator Framework Object
- Simple Monte Carlo Iterator
- Synchronized List Iterator

IsRequired Formula

Some modules will always require that a parameter be specified externally, and other modules will only need to have the parameter when certain options are operating. In the cases where the parameter is only required under certain operating schemes, the MIMS framework uses the IsRequired field.

When the IsRequired field evaluates to True, then the parameter is necessary for the module instance to execute. The evaluation expression in the IsRequired field may contain other parameters but should always evaluate to a logical value of true or false. Use Ctrl-C and Ctrl-V to cut and paste from other formula expressions.

Here are some formula examples:

Parameter Condition	IsRequired Formula Expression
Boolean = true	${val("bool1")}
Integer > 10	${val("int1") > 10}
Boolean and integer	${val("bool1") and val("int1")>10}
String = "abcdef"	${ str("str1")=="abcdef" }

To understand more about how to construct the formulas, see the section on Calling Parameters.

Iterator Framework Objects

The MIMS Iterator Framework Objects can be useful tools for sequencing multiple runs of:

- Sequential time periods (e.g., multiple simulations of weeks in a year)
- Different time periods (e.g., a time period in July 1995 and one in August 1997)
- Different emissions control strategies and a baseline
- Projections of control strategies out to multiple future years
- Different chemical mechanisms so that their output files can be compared
- Monte Carlo simulations
- Calibrations and optimizations

When a user creates an Iterator framework object, an object window appears. If the user clicks to the Processes tab in the object window, a process called Iterate appears with a module instance called Iterator. Users should double-click the mouse on the module instance to open a module instance window.

The module instance window will be titled Module Instance Iterator, and the Parameters menu lists two additional items not found in other object windows. The *Set Iterand...* menu item is used to set the module/scenario that the user wants to run repeatedly and also prompts the user for the parameters that should be passed in and out of the nested module/scenario. The *Show Module* menu item will display the associated module.

Below readers will find more details on the following MIMS iterators:

- TemporalIterator (operates differently than the other iterator objects)
- Simple Monte Carlo Iterator
- Simple Monte Carlo Iterator (max)
- Synchronized List Iterator

Histograms are often an effective way to display results from a Monte Carlo simulation, so MIMS provides a Histogram Plotter module.

The ***Simple Monte Carlo Iterator*** object can be used to send random numbers to a module/scenario for a particular parameter and track the effect on the output values. When users create a Simple Monte Carlo Iterator object with a time series output, MIMS will track the final value from the time series. To track the maximum value in the time series, users should employ the Simple Monte Carlo Iterator (Max).

To generate this iterator, create a Simple Monte Carlo Iterator in the project window by clicking on the New button. The framework object window will appear for the iterator. Click on the Processes tab. Highlight the Iterate module, and press the Edit Module Instance button to make the module instance window appear.

From the Module Instance Iterator window, click on "Set Iterand" in the Parameters menu. Choose the desired module/scenario as the member. An Inputs/Outputs dialog will appear. On the Inputs tab, select the variable you would like to randomize in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. On the Outputs tab, select the desired output parameter in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. Click the OK button. The dialog closes. Three new parameters are added to the parameters list in the module instance window:

- Number of Realizations,
- Distribution of 'input parameter,' and
- Realizations of 'output parameter.'

The number of realizations should be set to the desired number of iterations. Click the Set button beside the Distribution of 'input parameter.' Choose a uniform or normal distribution, and then assign a name and parameters to the distribution. After choosing the parameters, click the OK button.

The ***Simple Monte Carlo Iterator (Max)*** object can be used to send random numbers to a module/scenario for a particular parameter and track the effect on the maxima of the output values. When users create a Simple Monte Carlo Iterator (Max) object with a time series output, MIMS will track the maximum value from the time series.

To generate this iterator, create a Simple Monte Carlo Iterator (Max) in the project window by clicking on the New button. The framework object window will appear for the iterator. Click on the Processes tab. Highlight the Iterate module, and press the Edit Module Instance button to make the module instance window appear.

From the Module Instance Iterator window, click on "Set Iterand" in the Parameters menu. Choose the desired module/scenario as the member. An Inputs/Outputs dialog will appear. On the Inputs tab, select the variable you would like to randomize in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. On the Outputs tab, select the desired output parameter in the Unselected Items list and click Add.

The parameter will now be listed in the Selected Items List. Click the OK button. The dialog closes. Three new parameters are added to the parameters list in the module instance window:

- Number of Realizations,
- Distribution of 'input parameter,' and
- Realizations of 'output parameter.'

The number of realizations should be set to the desired number of iterations. Click the Set button beside the Distribution of 'input parameter.' Choose a uniform or normal distribution, and then assign a name and parameters to the distribution. After choosing the parameters, click the OK button.

Instead of iterating on random numbers, the ***Synchronized List Iterator*** will repeat a module using an input parameter whose values are taken from a list. The list could specify numbers, text strings, or even file names and directories. This framework object might be used to configure a series of model simulations that consider different cases. For example, each string might hold the name of a different chemical that should be considered in a hazardous waste project. For an air quality strategy design project, each string might hold the name of a control strategy approach (e.g., "Baseline," "Bold Reductions," and "Bold with Constraints").

To generate this iterator, create a Synchronized List Iterator in the project window by clicking on the New button. The framework object window will appear for the iterator. Click on the Processes tab. Highlight the Iterate module, and press the Edit Module Instance button to make the module instance window appear.

From the Module Instance Iterator window, click on "Set Iterand" in the Parameters menu. Choose the desired module/scenario as the member. An Inputs/Outputs dialog will appear. On the Inputs tab, select the variable you would like to randomize in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. On the Outputs tab, select the desired output parameter in the Unselected Items list and click Add. The parameter will now be listed in the Selected Items List. Click the OK button. The dialog closes. Four new parameters are added to the parameters list in the module instance window:

- Execution Method,
- Random Number Seed,
- list of 'input parameter,' and
- list of 'output parameter.'

Click on the Edit button beside the list of 'input parameter' variable to open a dialog called Edit List of Parameters Parameter Type. From this window, click on the add parameter button + to add variables to the list. Users may specify the value in this list but no other aspects of the input parameter. The reorder button ⇅ may be used to arrange the order of the synchronized list.

MIMS Project (presented in the Project Window)

A MIMS Project is the highest level that a user views within the MIMS framework and serves as a library of all of the information involved in a study. The MIMS Project holds all of the scenarios, domain objects, framework objects, modules, parameters, and other members that are used within a study. Within a given MIMS Project, all members are available for any scenario and do not have to be re-created.

The MIMS Project for a session is created or chosen from the Project Selection Window (first screen displayed when the MIMS program begins).

Module (presented in the Module Window)

A MIMS Module is a software program that the MIMS framework invokes. Each Module can only implement one MIMS Process and performs according to the module instance's Parameters specifications. A Module Instance describes a single invocation of a MIMS module, according to the Parameters specified in the MIMS module instance.

User-created MIMS modules can focus on a particular environmental media. The environmental media would be specifed as the module's type, and the module instance could then be included in domain objects of the same type. Sample modules (and their types) include:

- BASINS (surface water)
- CCTM (atmosphere-land)
- MM5 (atmosphere)
- PRSYM (groundwater)
- SMOKE (atmosphere)
- SWOOMS (surface water)
- TOPLATS (land surface)

To view the details about a module, users may access the Module Window through the Project Window or the Module Instance Window. As an example, from the Project Window, choose Default External Module as the category and then double click on the desired module. From the Module Instance Window, select the Parameters menu and pull down to *Show Module*. The Module Window will appear and describe the settings for this module.

Module Instance (presented in the Module Instance Window)

A module instance describes a single operation of a MIMS Module, according to the Parameters specified for that particular MIMS Process. To execute a Module Instance, connect all the necessary parameters so that a green open circle appears beside the process in the scenario window. Then choose Execute All from the Scenario menu in the Scenario window. After a module instance has successfully executed, a check mark appears beside the Process in the Scenario Window.

Parameter

A MIMS Parameter is any data, metadata, or placeholder that defines a system and determines (or limits) its performance. Within the MIMS Framework, parameters provide the input, output, and operational configuration of a module, domain object, scenario, or other member. Example parameters include:

- Integers and floating point numbers describing physical phenomena
- Files with input data
- Time steps
- File names for output data
- Geospatial data set
- Geographic projection
- Choice of chemical mechanism

Every MIMS parameter has a type associated with it, and most will have specified values. The following list shows some examples of the parameter types included in MIMS:

Parameter Type	Description
Boolean flag (logical)	True or false
Chemical mechanism	Table of chemical species and reaction equations
Date	Date-time value that includes year, month, day, hour, minute, and seconds
Directory path name	File directory on the system drive
Duration	Simulation time period or other length of time
FileName	Contains the name of the file and possibly path. This can be used to write an output file that will not exist until used.
File	Name for a file that already exists when it is called. The file name directs where the simulation should get input information or place output.
Floating point number	Any real number
Integer	Number whose absolute value can be expressed as a whole number
M3IO File Variable	Reference to a variable found within a M3IO file
Parameter Group	Set of parameters identified by a single group name
String	Text

Typed File	File name that will only allow choices that fall under a particular defined file tag

The parameter type is chosen when the parameter is created and cannot be altered. Parameters are associated with a particular MIMS member and may be viewed and set from the member's window. Within a MIMS member (e.g., scenario, domain object, or module), each parameter must have a unique name. The output parameter from one MIMS module instance or other member may serve as the input for another member through parameter connections.

Parameter Connection

Within the MIMS framework , the output Parameter from one module instance may serve as the input for another module instance, but only if the output is set at the object level. Two module instances within the same object will share parameters that are listed as domain object parameters. If a connection is set between two different objects, the parameters from the object's module instances will be passed. The user matches these inputs and outputs through Parameter Connections.

The Scenario Window, Domain Object Window, Framework Object Window, and the Module Instance Window allow users to make Parameter Connections. In the Scenario Window, the user right clicks on the object requiring input, chooses Add ParameterSource from the menu, and then clicks on the source object. In the other windows (e.g., module instance windows), the user selects the Parameter, clicks on the Plug Connection, and then selects the domain or framework object containing the Parameter.

Within the MIMS framework, users also need to know how scenario, object, and module parameters are associated. Input parameters pass from the scenario level to the module level, and the output parameters from a module are passed up to the scenario level.

Parameter Group

A MIMS Parameter Group provides a convenient method for organizing all of the parameters necessary to execute a MIMS Scenario. The parameter group stores parameters together and can be viewed on a single screen (Parameter Group Window). By grouping parameters, users can avoid scrolling through long lists of variables that may be associated with any given module. Parameter groups also make it easier to introduce a large number of parameters to multiple members within a MIMS Project.

Since parameters within a group are considered parameters of the member themselves, users must make sure that the parameter group's elements have unique names within the member.

Any MIMS Module will draw the parameters it needs from a Parameter Group and ignore the rest. This grouping approach also allows users to see which parameters are passed between the MIMS Domain Objects and MIMS Framework Objects on a single screen, but the modules do not treat individual parameters or parameters within a group differently.

Process

A MIMS process describes a single action within the MIMS framework. In most cases each process executes a single module instance with the specifications determined by the input parameters for the module instance. The MIMS module instance will gather the input Parameter values, operate on them using a MIMS Module, and export the data to the specified output Parameter files. This execution series of a MIMS process is referred to as a Module Instance.

One or more Processes may be contained in a MIMS Domain Object.

To locate a particular Process within the MIMS framework, go to the MIMS Scenario Window (the window with Tree and Graph views). Click on the "+" next to the Domain Object to see its list of processes (or double-click on the Domain Object name to view all of the processes and parameters associated with that Domain Object).

The process name appears before the colon, and the associated module name is shown after that. Right-click on the process to set, update, or delete the associated module instance. Users may also choose to assign a process to a domain object but not assign a module instance. In this case, computations will not be performed, but the process placeholder will describe the physical/chemical characteristics and operations of the domain object.

If you double-click on the Process in the Scenario Window (and a module instance is associated with the process), a new window appears. The new window is named Module Instance <*module name*> and allows the user to view the Module and the Parameter specifications.

Scenario (presented in the Scenario Window)

A Scenario represents a complete simulation and holds all of the information necessary to reproduce the simulation. A scenario may start by creating a model's input files, then execute different models, and end with the analysis of the output. The Scenario is the second hierarchical level in MIMS; most MIMS Projects will be composed of one or more Scenarios.

A MIMS Scenario arranges all of the relationships among the MIMS building blocks (MIMS Domain Objects, MIMS Framework Objects, MIMS Processes, MIMS Modules, and Parameters) so they can be executed. To choose a MIMS Scenario, look under the Project Window. Under the Category menu, select Scenario to view the available scenarios.

Users should save each scenario from which they plan to gather results. Then they should duplicate and modify the copy of that scenario when they wish to change any element:

- framework object (e.g., MIMS Iterator),
- domain object,
- process,
- module, or
- parameter.

The user can use the Duplicate button in the Project Window to copy scenarios.

A scenario may also behave as a module within another scenario. Under the Scenario menu of the Scenario Window for the nested scenario, choose the Edit Parameter Values option. A dialog will appear. Set the object type and process name to the values that will be used in the outer scenario. When you set a module in the outer scenario, the list of choices will now include the scenario you wish to nest. Remember that the only parameters that can be passed into and out of the nested scenario are the ones specified as scenario parameters (for the nested scenario).

Status Field

The Status field is shown for all Parameters in the Module Window, Module Instance Window, Domain Object Window, Framework Object Window, and Scenario Parameters Window. It indicates how the parameter should be used during a module instance. Most of the parameter status fields (in, out, inout, locked, local, and required) are set when the parameter is created using the Add Parameter function. After this, the input/output/local status and locked status of a variable cannot be altered.

The table below gives a brief description of the status conditions and how to set them:

Set by...	Function	Appears as...
Is Input	Used as input during the module instance	in
Is Local	When a scenario is used as a module, the local scenario parameters are not visible or accessible outside this module	local
Is Output	Created as output during the execution of the module instance	out
Is Required	Necessary for successful completion of the module instance	required
Is Input and Output	Both used as input and created as output during the execution of a module instance	inout
Is ReadOnly	May be altered by modules or a connection to another parameter but not directly by the user	locked

Parameter connection	Indicates that another module instance or domain object uses the parameter's value	used
Design of modules	Ignores the parameter names and domain object types when matching parameters. This setting is commonly used for plotting tools that operate on many parameters.	*

Temporal Iterator

The TemporalIterator framework object can provide time stepping for modules that do not have their own time stepping devices or when an additional level of time stepping is required. Examples include the following:

- A set of models that step forward in time together might be incorporated into the MIMS framework by allowing the TemporalIterator to control the time stepping for all of the models instead of trying to coordinate the time step loops in each model separately.
- A multiple-day air quality episode might be divided into one-day segments to reduce file sizes. In such a case, even though the air quality model has its own internal time steps, it must be restarted for each day in the episode. The TemporalIterator could initiate the daily restart.

The inputs to the TemporalIterator are the *starting time*, the *duration* of the iterator's time step, and the *number of time steps* to consider. For instance, a starting time of 1:00 pm, a time step of one hour, and two steps would invoke the dependent models for times corresponding to 1:00 and 2:00 pm.

For a module to use the iterator's output value, the user must establish a Parameter Connection to the temporal iterator's output parameter and include this variable in the input to at least one module instance.

Other standard iterators are described in the section on Iterator Framework Objects.

How to Use MIMS

Calling Parameters

The MIMS Framework uses the Python code format for specifying parameters that should be passed between objects, modules, and scenarios. Under this format, users specify that a parameter should be called by including the parameter name in single or double quotation marks. For example, if the **numeric** parameter *nlayers* has the number 12 associated with it, a user would type "nlayers" or 'nlayers' as the parameter value in the module window's parameter table. When the module needs to know the number of layers (nlayers), the formula "nlayers" will be evaluated as 12.

Similarly, for a **Boolean** parameter called *test*, the expression "test" would evaluate to be either true or false. By right-clicking on the expression in the parameters table, users can evaluate the value currently assigned to a parameter.

Note that the Python code format is case-sensitive, so "nlayers" and "Nlayers" would be treated as two distinct parameters.

For **string**, **date/time**, and **environment** variables, MIMS users must specify formulas to read the parameters. A formula is denoted by a *${* on the left and *}* on the right. If *pstring* is a variable with value "abcdef", then MIMS users would call this parameter with the expression `${str("pstring")}`. MIMS also uses the following formulas to evaluate strings:

String Parameter	Evaluation
${str("pstring")}	abcdef
${str("pstring") [2:3]}	cde (2 specifies the character to begin the substring after and 3 shows the number of characters in the substring)
${str("pstring").upper()}	ABCDEF (changes strings to all upper case characters)
${str("pstring").lower()}	abcdef (changes strings to all lower case characters)
${str("pstring")+str("pstring") [2:3]}	abcdefcde (the + sign concatenates strings and substrings)

Similarly, a date/time variable and its components can be evaluated with formulas. The table below shows evaluations of the *pdate* variable (set to 07/14/2003 10:50:06):

Date/Time Parameter	Evaluation
${getDate("pdate","yyyymmdd")}	20030714
${getDate("pdate","mmddyyyy")}	07142003
${getDay("pdate")}	14
${getJDay("pdate")}	195 (Julian day in the calendar year)
${getYear("pdate")}	2003 (returns the four-digit year)
${getYear2("pdate")}	03 (returns the two-digit year)
${getHour("pdate")}	10
${getMinute("pdate")}	50
${getSeconds("pdate")}	06

Environment variables (such as the operating system *OS*) can be called with a simple formula like `${getEnv("OS")}`.

MIMS also allows users to specify a parameter from two choices with **ifElse** formulas. The formula `${ifElse("test", getYear("pdate"), getYear2("pdate")}` would return 2003 if the *test* variable is true and 03 if the *test* variable is false. The ifElse formulas are similar to those used for IsRequired formulas.

To understand more about constructing formulas to operate in the MIMS framework, visit the Python documentation web site.

Chemical Mechanism Family Editor

This window describes the available Chemical Mechanisms available within a particular Chemical Mechanism Family. The chemical mechanism family is a parameter associated with MIMS Modules.

A list shows some key characteristics of the chemical mechanisms within the family: whether or not aerosol chemistry is included, whether or not aqueous phase chemistry is included, and whether or not the mechanisms can be altered (ReadOnly). The first two characteristics are important for atmospheric chemistry. If the Aerosol or Aqueous boxes are checked, it indicates that at least one chemical species in the mechanism is associated with the aerosol or aqueous phases. Turning off these check boxes will ignore the reactions/conversions associated with the Aerosol and Aqueous phases. If no associations exist with these phases within the chemical mechanism, the box cannot be checked.

One chemical mechanism name listed in the window will appear in black, while all others appear in gray. The black name indicates the Active Chemical Mechanism, the one that will currently be used in Module Instances. To select a different Active Chemical Mechanism, click on the new mechanism and then on the Set Active button.

This window also contains menu options and buttons to perform the following tasks:

Identifier	Function	Window Access	Menu Access
Add Mechanism	Adds a new chemical mechanism to the list, starting with a copy of the Active Chemical Mechanism	▭	Edit
Close Chemical Mechanism	Closes this window but not necessarily the MIMS framework		File
Delete Mechanism	Permanently erases the selected chemical mechanism	▭	Edit

Edit Description	Allows the user to describe the selected chemical mechanism. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▭	Edit
Edit Mechanism	Opens the Edit Dialog for Mechanism for the selected chemical mechanism, allowing the user to view and alter the reactions, reaction rates, and species considered	▭	Edit
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Export Mechanism	Saves the selected chemical mechanism and all of its associated parameters to a file		Edit
Import Mechanism	Opens a file to add new chemical mechanisms to the current chemical mechanism family	▭	Edit
Open Project...	Opens an existing Project and the associated Project Window	Ctrl-O	File
Project Selection Window	Brings the list of Projects into the foreground		File
Set Active	Marks the selected chemical mechanism in black and uses that one in subsequent module instances	▭	Edit

Domain Object Window

The Domain Object Window (titled "Domain Object"+domain object name) lets you view the MIMS Processes and Parameters associated with the MIMS Domain Object.

The type field is used to determine which modules and parameters may be connected with this domain object. For example, a domain object with a "surface water" type could be used to describe domain objects for both Lake Erie and the Mississippi River. Type names are case sensitive.

Click the tabs to view a list of either the Parameters or Processes. The parameters tab lists the Parameters that are associated with the Domain Object and their origins. The list contains the following information:

- Parameter name
- Value (output value or file name where information is stored)
- Source (MIMS Module or other member that creates the parameter)
- Status (in, out, inout, required, used, local, *, and locked)

The specifications about domain object type, the domain object parameters, and the values can be set from the domain object window opened from the Project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when the user opens the domain object window from within the scenario, but the other specifications may not be altered.

The Processes tab lists the MIMS Processes (and associated Modules) for the Domain Object. Double-clicking on a Process will call up a Module Instance Window for the Process.

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines. The feature only works for domain object windows opened from the project window.	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window		Parameters
Add Parameter Values	Copies parameters from the contained module into the domain object	▭	
Add Process	Adds a new MIMS Process to the Domain Object	▭	
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters Window. See how parameters are associated.	⊞	Parameters
Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.		Parameters
Clear Module Instance	Removes all Parameter values and Modules from the highlighted MIMS Process	▭	
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters
Close Domain Object	Closes the Domain Object window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)	↕	Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from another MIMS Domain Object or MIMS Framework Object. See how parameters are associated		Parameters

	associated.		
Copy	Duplicates a Parameter found in the Domain Object Window (and allows it to be placed in another Domain Object Window or Framework Object Window by the Paste command)		Edit
Copy from Module	Brings up a list of modules and imports the parameters of the selected modules into the parameter list. See how parameters are associated.		Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type		Parameters
Cut	Erases a Parameter from the Domain Object Window (and allows it to be placed in another Domain Object Window or Framework Object Window by the Paste command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Edit
Edit Description	Allows the user to describe the Domain Object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit Description	Allows the user to describe the highlighted Process. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req*	Parameters
Edit Module Instance	Opens a Module Instance Window for the highlighted Process		
Edit Parameter Value Description	Allows the user to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Parameters

Edit Parameter Type	Edits the description of the highlighted parameter. In some cases, additional features of the parameter type can be altered.	✎	Parameters
Evaluate Parameter	Displays the current value for the highlighted parameter	Right click	
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Import Parameter Values	Prompts user to select a file containing the parameter values		File
Open Project...	Opens an existing Project and the associated Project Window	Ctrl-O	File
Paste	Inserts a Parameter that has been cut or copied into the Module Instance Window	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Remove Processes	Disconnects the selected processes from the domain object		
Reorder parameter	Moves the selected parameter one place higher in the list		Parameters
Save Domain Object	Saves the Domain Object		File
Set From File	Generates a set of M3IO file variable type parameters by reading the specified file	📂	Parameters

Set Icon	Assign an image to the particular domain object		Edit
Set Module Instance	Attaches a module to the highlighted process	▭	
Set Parameter From Project	Assigns the value from this parameter at the Project level to the value of this parameter	↥	Parameters
Undo	Cancels the previous action	↺	Edit
View File	Opens a browser showing the file contents for the selected parameter	Right click	

Edit Administration Information Window

This window is composed of two tab listings: Computers and Viewers. The Computers tab allows users to specify the names, operating systems, and number of processors available to MIMS.

The Viewers tab allows users to set the program with which to browse file contents. The viewer that MIMS chooses depends on the file extensions or file tags (for tagged file parameters) as specified on the Viewers tab. The View File command is generally available when a user right clicks the mouse on a file parameter.

Use the Edit Menu to insert and delete rows. To enable some basic viewers, set one row to read "txt; log; xml" under the File Extensions heading and "default" under the Viewer Command heading. This will instruct the View File command to open files with extensions txt, log, and xml to open under "notepad %" for Microsoft Windows and a Java viewer method for UNIX systems.

To enable viewing additional files, insert a second row. Type "nc; m3io; ioapi" under the File Extensions heading and "java ioapifileviewer.session.Viewer %s" under the Viewer Command heading.

Edit Dialog for Mechanism

When a user selects the Edit button from the Chemical Mechanism Family Editor, the Edit Dialog for Mechanism window appears. Currently this window supports a standard format of menus aimed at air quality modules; you may click through a series of tabs to view five different lists:

- Reactions – list of all chemical reactions (and phase conversions) and their kinetic rate constants
- Gas Species – list of each chemical species (or grouped species) found in the gas phase, its known chemical and physical properties, and associated variables
- Aerosol Species – same as Gas Species list, for aerosol components (species must be listed here to have the Aerosol box checked on the Chemical Mechanism Family Editor
- Non-Reactive Species – same as Gas Species list, for non-reactive components
- Tracer Species – list of chemical species that will be tracked from the emissions source to the receptor location

To see what each of the fields stands for on these lists, see the List of Chemical Mechanism Parameters.

Most of this window is edited directly by double-clicking on the cell that you want to change. A few additional operations are available through the menus:

Identifier	Function	Menu Access
Add Surrogate	Adds a surrogate species and its conversion factor to allow tracking of measurable quantities	Edit
Add Switch	Adds a switch to allow users to leave a species out of calculations	Edit

Close with saving	Closes this window but not necessarily the MIMS framework; saves any changes made to the Chemical Mechanism unless the mechanism was marked ReadOnly in the Chemical Mechanism Family Editor	File
Close without saving	Closes this window but not necessarily the MIMS framework; does not save any changes made to the Chemical Mechanism	File
Delete Row(s)	Permanently removes the selected rows from the list	Edit
Delete Surrogate	Removes the surrogate species	Edit
Delete Switch	Removes the switch to allow users to leave a species out of calculations	Edit
Insert Row	Adds a new row to the front table for new reactions or chemical species	Edit
Print	Sends the front table list to the printer	File

Framework Object Window

The Framework Object Window (titled "Framework Object"+framework object name) lets you view the MIMS Processes and Parameters associated with the MIMS Framework Object. Use the Type field to describe the modules and parameters associated with this Framework Object. Click the tabs to view a list of either the Parameters or Processes.

The parameters tab lists the Parameters that are output from the Framework Object and their origins. The list contains the following information:

- Parameter name
- Annotation (more detailed description of the parameter)
- Value (output value or file name where information is stored)
- Source (MIMS Module that creates the parameter)
- Status (in, out, inout, required, used, local, *, and locked)

The specifications about framework object type, the framework object parameters, and the values can be set from the framework object window opened from the Project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when the user opens the framework object window from within the scenario, but the other specifications may not be altered.

The processes tab lists the MIMS Processes (and associated Modules) for the Framework Object. Double-clicking on a Process will call up an Module Instance Window for the Process.

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines. The feature only works for framework object windows opened from the project window.	+	Parameters

	project window.		
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window		Parameters
Add Parameter Values	Copies parameters from the contained module into the framework object		
Add Process	Adds a new MIMS Process to the framework object		
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters Window. See how parameters are associated.	⊞	Parameters
Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.		Parameters
Clear Module Instance	Removes all Parameter values and Modules from the highlighted MIMS Process		
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters
Close Domain Object	Closes the Framework Object Window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)		Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from another MIMS Domain Object or MIMS Framework Object. See how parameters are associated.		Parameters
Copy	Duplicates a Parameter found in the Framework Object Window (and allows it to be placed in another Domain Object Window or Framework Object Window by the Paste command)		Edit

	Object Window by the Paste command)		
Copy from Module	Brings up a list of modules and imports the parameters of the selected modules into the parameter list. See how parameters are associated.		Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type		Parameters
Cut	Erases a Parameter from the Framework Object Window (and allows it to be placed in another Domain Object Window or Framework Object Window by the Paste command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Edit
Edit Description	Allows the user to describe the Framework Object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit Description	Allows the user to describe the highlighted Process. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Module Instance	Opens an Module Instance Window for the highlighted Process		
Edit Parameter Value Description	Allows the user to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Parameters
Edit Parameter Type	Edits the description of the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Parameters

Evaluate Parameter	Displays the current value for the highlighted parameter	Right click	
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Import Parameter Values	Prompts user to select a file containing the parameter values		File
Open Project...	Opens an existing Project and the associated Project Window	Ctrl-O	File
Paste	Inserts a Parameter that has been cut or copied into the Framework Object Window	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Remove Processes	Disconnects the selected processes from the framework object	▭	
Reorder parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Save Framework Object	Saves the Framework Object		File
Set From File	Generates a set of M3IO file variable type parameters by reading the specified file	📂	Parameters
Set Icon	Assign an image to the particular framework object		Edit

Set Module Instance	Attaches a module to the highlighted process	▭	
Set Parameter From Project	Assigns the value from this parameter at the Project level to the value of this parameter		Parameters
Undo	Cancels the previous action	↺	Edit
View File	Opens a browser showing the file contents for the selected parameter	Right click	

Grid Family Window

When a user chooses to view a Grid Family or create a new Grid Family from the Project Window, the Grid Family GUI Window appears. This window can be used to create grids, visualize how multiple grids overlay land areas, and print the associated maps. If you use nested grids, this type of interface will allow you to produce and display the grid boundaries quickly.

If you choose to create a new Grid Family, the window initially shows a map on the right with North American states and provinces (except Hawaii) with longitude/latitude coordinate system. The window is divided into three sections: the Grid Chooser panel, the Grid Info panel, and the Grid Display panel. The Grid Chooser panel (upper left) allows users to specify the coordinate system. The Grid Info panel (lower left) lets the user know the exact coordinates of the grid boundaries, and the Grid Display panel (right side) overlays the grids on familiar land areas (e.g., states, provinces, counties, and cities).

To create a nested grid within this interface,

1. Press the Edit button in the Grid Chooser panel.
2. Select the name and map coordinate system. Then press OK.
3. Click the Add New button in the Grid Chooser panel and provide the name for the coarse grid.
4. In the Grid Info panel (lower left), type in the lower left and upper right coordinates for the coarse grid. Then type the numbers of cells in the x and y directions.
5. Click on the Calc button beside DxDy to create the coarse grid. The coarse grid will be displayed in blue in the Grid Display panel.
6. Click the Add New button in the Grid Chooser panel and provide the name for the fine grid.
7. In the Grid Info panel (lower left), type in the lower left and upper right coordinates for the fine grid. Then type the numbers of cells in the x and y directions.
8. Click on the Calc button beside DxDy to create the fine grid. The fine grid will be displayed in green in the Grid Display panel.

The actions of the menu options and screen buttons are described below:

Identifier	Interface	Function
Add New	Button	Places a new grid into the current grid family
Calc (beside Lower Left)	Button	Calculates a grid layer based on the information in the upper right, number of cells, and delta X and Y fields
Calc (beside Upper Right)	Button	Calculates a grid layer based on the information in the lower left, number of cells, and delta X and Y fields
Calc (beside Num Cells)	Button	Calculates a grid layer based on the information in the lower left, upper right, and delta X and Y fields
Calc (beside Delta X,Y)	Button	Calculates a grid layer based on the information in the lower left, upper right, and number of cells fields
Close Grid Family	File Menu	Closes this window but not necessarily the MIMS framework
Delete	Button	Permanently removes a grid layer
Duplicate	Button	Creates a copy of the existing grid layer
Edit	Button	Allows the user to change the name of the grid family or the coordinate system
Edit Description	Button	Adds a paragraph describing the selected grid layer
Edit Layers	Edit Menu/ Layers Button	Allows users to determine which ESRI shape files are shown with the grids, add new layers, and choose which layer to place on top
Exit MIMS Framework	File Menu	Closes all windows and programs associated with the MIMS framework
Export GRIDDESC	File Menu	Saves an existing grid family to a file location
Import GRIDDESC	File Menu	Loads an existing grid family from a file
Nest Grid	Button	Subdivide a portion of one grid into smaller subunits
Open Overlay Files	File Menu	Opens a text file…???

Print	Button	Prints the map displayed in the Grid Display panel
Redo	Edit Menu/ Ctrl-Y	Repeat the previous action
Rename	Button	Change the name of a grid layer
ROI	Button	Lets the user directly enter the minimum and maximum latitudes and longitudes for the region of interest (ROI) that will be shown in the display
Set LL&UR	Button	Drag the mouse to set the lower left and upper right coordinates of the grid
Show Grid	Check Box	Determines whether or not the grid layer is shown in the Grid Display panel
Thin Lines By	Field Entry	???
Undo	Edit Menu/ Ctrl-Z	Cancel the previous action
XY/LL	Check boxes	Determines if the X/Y or latitude/longitude coordinates for the mouse are displayed in the Grid Display Panel
Zoom Area	Button	Allows users to drag the mouse across the desired zoom area
Zoom In	🔍+	Changes the zoom on the Grid Display panel to double the magnification
Zoom Out	🔍−	Changes the zoom on the Grid Display panel to halve the magnification

Module Window

The Module Window (titled "Module"+module name) lists the MIMS Processes, Parameters, and the exterior program associated with the MIMS Module. In addition to the Parameters list, the Module Window presents:

- Execution Command (the path and name for the software program as well as any command line arguments)
- Control files (viewer to the files read by an external module)
- Compatible Domain Object Type (the domain object type that may contain the module, not the name of the domain object)
- Name of Process This Implements (MIMS Process associated with the Module)

Users may type directly into these fields or select Browse when necessary. A checkbox with "If no output, do not wait for completion" is also available to control processing. The Perform Preprocess option allows preliminary execution of arbitrary script code (e.g., to perform consistency checks among parameters).

The Parameters that are input and output from the MIMS Module are also listed in this window. The list contains the following information:

- Object Type (MIMS Domain Object or Framework Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Internal name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Status (in, out, inout, required, used, local, *, and locked)

If opened from a Scenario Window, the parameters cannot be added, altered, or removed from the Module Window. They may be adjusted if the Module Window is opened from the Project Window.

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window	⊞	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters
Close Module	Closes the Module Window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)	↕	Parameters
Copy	Duplicates a Parameter found in the Domain Object Window (and allows it to be placed in another Domain Object Window or Framework Object Window by the Paste command)	⎘	Edit
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type	⤓	Parameters
Cut	Erases a Parameter from the Module Window (and allows it to be placed in another member by the Paste command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Edit
Edit Description	Allows the user to describe the module. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit

Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Parameter Value Description	Allows the user to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	🗎	Parameters
Edit Parameter Type	Edits the units and description of the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Export	Saves the module and all of its associated parameters to a file		File
Open Project...	Opens an existing Project and the associated Project Window	Ctrl-O	File
Paste	Inserts a Parameter that has been cut or copied into the Module Window	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Reorder parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file	📂	Parameters

Set Parameter from Project	Assigns the value from this parameter at the Project level to the value of this parameter		Parameters
Undo	Cancels the previous action	↶	Edit

Module Instance Window

The Module Instance Window (titled "Module Instance"+Module name) lists the parameters associated with a MIMS Module. The module tab shows the same module description lines seen on the Module Window (execution Command, Control Files, Compatible Domain Object Type, Process Name, Description, and instructions to wait for completion). The Parameters that are input and output during the execution of the Module Instance are listed in this window. The list contains the following information:

- Object Type (MIMS Domain Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Source (if the parameter is read from a domain or framework object output, this field indicates the object name)
- Status (in, out, inout, required, used, local, *, and locked)

If the window is opened from the Scenario Window, parameters cannot be added, altered, or removed from the Module Instance Window. However, users can change parameters in Module Windows that were accessed through the Project Window.

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines	+	Parameters
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters Window. See how parameters are associated.	⊞	Parameters

Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.		Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter		Parameters
Close Module Instance	Closes the Module Instance Window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)		Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from a MIMS domain object or framework object. See how parameters are associated.		Parameters
Copy	Duplicates a Parameter found in the Module Instance Window (and allows it to be placed in another window by the Paste command)		Edit
Copy Parameter from Object	Adds a new parameter to the list, one created in a domain object window or framework object window		Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type		Parameters
Cut	Erases a Parameter from the Module Instance Window (and allows it to be placed in another member by the Paste command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Parameters
Edit Description	Allows the user to describe the module instance. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit

Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req*	Parameters
Edit Parameter Type	Edits the units and description of the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Evaluate Parameter	Determines the value associated with the highlighted parameter	Right click	
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Open Project...	Opens an existing Project and the associated Project Window	Ctrl-O	File
Paste	Inserts a Parameter that has been cut or copied into the Module Instance Window	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Reorder Parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file	📂	Parameters
Set Parameter from Project	Assigns the value from this parameter at the Project level to the value of this parameter	↑	Parameters

Show Module	Opens a Module Window for the module associated with this module instance. Parameters cannot be added or removed in the displayed module window.	□	Parameters
Undo	Cancels the previous action	↺	Edit
View File	Opens a browser showing the file contents for the selected parameter	Right click	
View Parameter Value Description	Allows the user to describe the value assigned to the highlighted parameter	▤	Parameters

Parameter Group Window

If you choose to employ a Parameter Group to define a whole set of Parameters under a single name, then click the Edit box in the row beside the parameter name. This will open up a window titled "Parameter Group of "+<parameter group name> with a list of the parameters contained within the group.

The list contains the following information:

- Object Type (the MIMS domain object or framework object type that may contain the parameter)
- Parameter (lists the name of the parameter)
- Internal Name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Status (in, out, inout, required, used, local, *, and locked)

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Accesss	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window		Parameters
Add to Group	Pulls individual parameters for the module, object, or scenario that have already been defined into the parameter group	⊕	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters

Close Parameter Group	Closes this window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)		Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type		Parameters
Delete Parameter	Deletes the selected parameters	✕	Edit
Edit Description	Allows the user to describe the Parameter Group. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Parameter Value Description	Edits the description for the highlighted parameter. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Parameters
Edit Parameter Type	Edits the description for the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Open Project…	Opens an existing Project and the associated Project Window	Ctrl-O	File
Print	Sends the Parameters Table to the printer		Parameters

Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻ Ctrl-Y	Edit
Remove from Group	Moves the selected group parameter out of the group and up to individual parameter status	⊖	Parameters
Reorder Parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file	📂	Parameters
Set Parameter from Project	Copies a parameter value from a project	↑	Parameters
Undo	Cancels the previous action	↺ Ctrl-Z	Edit

Project Window

The Project Window (titled with the Project name) lets you view all of the members available in your MIMS Project. Just select the type of member (e.g., Chemical Mechanism Family, Domain Object, Framework Object, Grid Family, Module, or Scenario) you want to see in more detail from the Category pull-down menu.

Double-click on the Project member you want to see in more detail, and a new window will be called up to describe that member.

The actions of the menu options and the screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add Temporal Controller	Brings a temporal Iterator Framework Object into the current project		File
Close Project	Closes this window but not the MIMS framework (unless this was the last open MIMS window)		File
Delete	Erases the selected member(s)	▭	
Edit Administration Information	Opens a window allowing system administrators to assign processors and viewers		File
Edit Project Description	Allows the users to describe the current study and its members. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit

Edit Recommended Domain Object Types	Allows users to create a preferred names list for object types		Edit
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Export Objects...	Creates a file that other MIMS Projects can access. The file describes members such as scenarios, domain objects, framework objects, modules, or processes		File
Import Objects	Prompts the user to select existing members to be included in the current project		File
Load User Defined Members	Allows users to import new types of members (classes) by specifying their file locations		File
New	Creates a new member within the current project	▭	
New Project...	Creates a new project and the associated project window		File
Open	Opens a window describing the selected member	▭	
Open Project...	Opens an existing project and the associated project window	Ctrl-O	File
Project Selection Window	Brings the list of projects into the foreground		File

Redo	Repeats previous action	Ctrl-Y	Edit
Remove User Defined Member	Removes the selected user defined member from the types of members		File
Rename...	Assigns a new name to the selected member	▭	
Undo	Cancels previous action	Ctrl-Z	Edit

Project Selection Window

This window displays the list of MIMS Projects that can be accessed by the MIMS framework . Double-click on a Project name to call up a Project Window.

The File Menu on this window provides ways to accomplish common file management tasks:

- New Project – creates a new Project from scratch
- Open Project – after selecting a Project, this option calls up the Project Window
- Rename Project... – gives a new name to the selected Project
- Delete Project – permanently removes a Project
- Import Project – brings new files into the MIMS framework as Projects
- Export Project... – creates a file that describes the Project in XML or binary format
- Close – closes the Opening MIMS Window
- Edit Administration Information – opens a window allowing system administrators to assign processors and viewers
- Exit MIMS Framework – closes all windows and exits the MIMS framework program

Scenario Window

The Scenario Window (titled "Scenario"+scenario name) lets you view all of the MIMS Domain Objects, MIMS Framework Objects, and MIMS Processes in your MIMS Scenario in two different formats. The left portion of the window provides a hierarchical tree view of the scenario; the right portion shows interconnected boxes in the graph view. The View Menu controls which views are shown, and their sizes may be adjusted by dragging the divider bar. Both views show the same information.

If the user clicks on Parameter Table under the View Menu, a list of scenario or member parameters appears within a separate view at the bottom of the Scenario Window. The Parameter Table view displays the parameters for the currently selected member.

To view the MIMS Processes for a Domain Object or Framework Object, click on the "+" beside the object name; to hide the MIMS Processes, click on the "-" symbol.

Each MIMS Process name is followed by the name of the associated MIMS Module Instance. The Process name is preceded by a circle to indicate the Execution status:

No symbol -- *no module instance has been connected to the process yet*

◌ **Missing inputs** -- *some required input parameters have not been set*

○ **Ready to process** -- *all required input parameters have values or are connected to the output of another execution*

● **Active** -- *the module instance is currently executing*

✓ **Successful** -- *the event completed successfully*

⊗ **Unsuccessful** -- *the event failed to complete*

The actions of the menu options and right clicks on the mouse are described below:

Identifier	Function	Window Access	Menu Access
Add Domain Object	Adds one of the project's MIMS Domain Objects to the Scenario	Right click	Edit
Add Framework Object	Adds one of the project's MIMS Framework Objects to the Scenario	Right click	Edit
Add Parameter Source	Builds a parameter connection (graph view only). Right click on the receiving domain object, choose Add Parameter Source, then left-click on the source domain object. See how parameters are associated.	Right click	
Add Process	Inserts a new process within the selected object	Right click	Object
Allow Use as Module	If users fill in the object type and process name fields within this window, the entire scenario will operate as a module within the current project.		Scenario
Clear Module Instance	Disconnects the selected MIMS module instance from the process	Right click	Object
Close	Closes this window but not necessarily the MIMS framework		File
Collapse All	Hides the MIMS processes associated with all of the domain and framework objects		Object
Copy Object to Project	Creates a duplicate of the selected object in the project under a new name		Object
Delete Object	Erases the selected domain or framework object from the scenario	Delete key or right click	Edit

Edit Administration Information	Opens a window allowing system administrators to assign processors and viewers		File
Edit Object	Opens a domain object window for the highlighted domain or framework object	Right click	Edit
Edit Parameter Values	Opens the Scenario Parameters Window and allows users to specify variables such as default paths and module instance protocols. This option cannot be selected when the Parameter Table view is shown (adjust from View menu).		Scenario
Execute All	Will execute all module instances in the scenario. **CAUTION:** Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust.	Ctrl-E	Scenario
Execute Selected	Will execute only the module instances that are currently highlighted in the scenario. **CAUTION:** Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust.		Scenario
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Expand All	Shows the MIMS processes associated with all of the domain and framework objects		Object
New Scenario	Creates a new scenario and the associated scenario window		File
Object Description	Allows the user to describe the highlighted domain or framework object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Object

Open Scenario	Opens an existing scenario and the associated scenario window		File
Print...	Sends the tree view, graph view, or parameters table to the printer		File
Project Selection Window	Brings the list of projects into the foreground		File
Redo	Repeats the previous action	Ctrl-Y	Edit
Remove Parameter Value Source	Removes a parameter connection	Right click	
Remove Process	Disconnects the selected MIMS process from the domain or framework object	Right click	Object
Rename Object	Assigns a new name to the highlighted domain or framework object	Right click	Object
Reset All	Returns all processes that have been executed to a ready-to-execute state	Ctrl-R	Scenario
Reset Selected	Returns the highlighted MIMS processes to a ready-to-execute state		Scenario
Save	Saves the scenario within the project. Even if the Save button is not clicked, open scenarios are saved when users exit MIMS.	Ctrl-S	File
Scenario Description	Allows the user to describe the scenario. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Scenario
Set Module Instance	Connects a MIMS module instance to the selected process	Right click	Object

Set Node View Size	Changes the font size in both the tree and graph views	Right click	View
Stop All	Kills the execution of all MIMS processes that began with the Execute All or Execute Selected commands	Ctrl-K	Scenario
Stop Selected	Kills the execution of the MIMS processes that began with the Execute Selected command		Scenario
Undo	Cancels the previous action	Ctrl-Z	Edit
Update Module Instance	Retrieves the updated version of the highlighted MIMS module from the project	Right click	Object
View Error Log	Opens a text editor showing the error logs for the MIMS module instance	Right click	
View Execution Log	Opens the file that recorded starts and completions of the module instances		Scenario
View Output Log	Opens a text editor showing the output logs for the MIMS module instance	Right click	
View Run Script	Opens a text editor showing the batch script files for the MIMS module instance	Right click	

Scenario Parameters Window

The Scenario Parameters Window (titled " Parameter Values of Scenario"+scenario name) lists the Parameters associated with a MIMS Scenario. The parameters listed here, such as Default Directory, apply to all of the members of the scenario. By default, the Scenario Parameters window shows the Default Directory, Execution Path, and whether or not old output files should be deleted before execution of the scenario's processes. The list contains the following information:

- Parameter (lists the name of the parameter)
- Value (parameter value or file name where information is stored)
- Source (if the parameter is read from a domain or framework object, this field indicates the object name)
- Status (in, out, inout, required, used, local, *, and locked)

MIMS output files and error logs are directed to the Default Directory. If no Default Directory is specified, the output files and error logs are stored in the directory from where MIMS was launched.

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window	▥+	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters

Close Parameter Values Table	Closes this window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)	↕	Parameters
Copy	Duplicates a Parameter found in the Scenario Parameters Window (and allows it to be placed in parameters tables by the Paste command)	⧉	Edit
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type	⤓	Parameters
Cut	Erases a Parameter found in the Scenario Parameters Window (and allows it to be placed in parameters tables by the Paste command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✗	Edit
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req*	Parameters
Edit Parameter Value Description	Edits the description for the highlighted parameter. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters
Edit Parameter Type	Edits the description for the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Open Project…	Opens an existing Project and the associated Project Window	Ctrl-O	File

Paste	Inserts a parameter that has been cut or copied into the parameters list		Edit
Print	Sends the Parameters Table to the printer		Parameters
Project Selection Window	Brings the list of Projects into the foreground		File
Redo	Repeats the previous action	↻ Ctrl-Y	Edit
Reorder Parameter	Moves the selected parameter one place higher in the list		Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file		Parameters
Set Parameter from Project	Copies a parameter value from a project		Parameters
Undo	Cancels the previous action	↺ Ctrl-Z	Edit

Appendix

Requesting Support

If you have questions concerning the use of the MIMS framework or spatial allocator with SMOKE or CMAQ, the Community Modeling and Analysis System Center provides support for those issues. Please submit your question to the CMAS Help Desk.

Other questions may be submitted or viewed on the MIMS support web page. The MIMS team will answer those questions as time permits.

Included Databases

List of databases used to be provided by the EPA

Licensing Information

When users install MIMS, a folder named Licenses is created. Within that folder, the software licenses appear.

Known Bugs

To view the current framework bugs, go to

https://sourceforge.net/tracker/?group_id=27492&atid=390709

Index